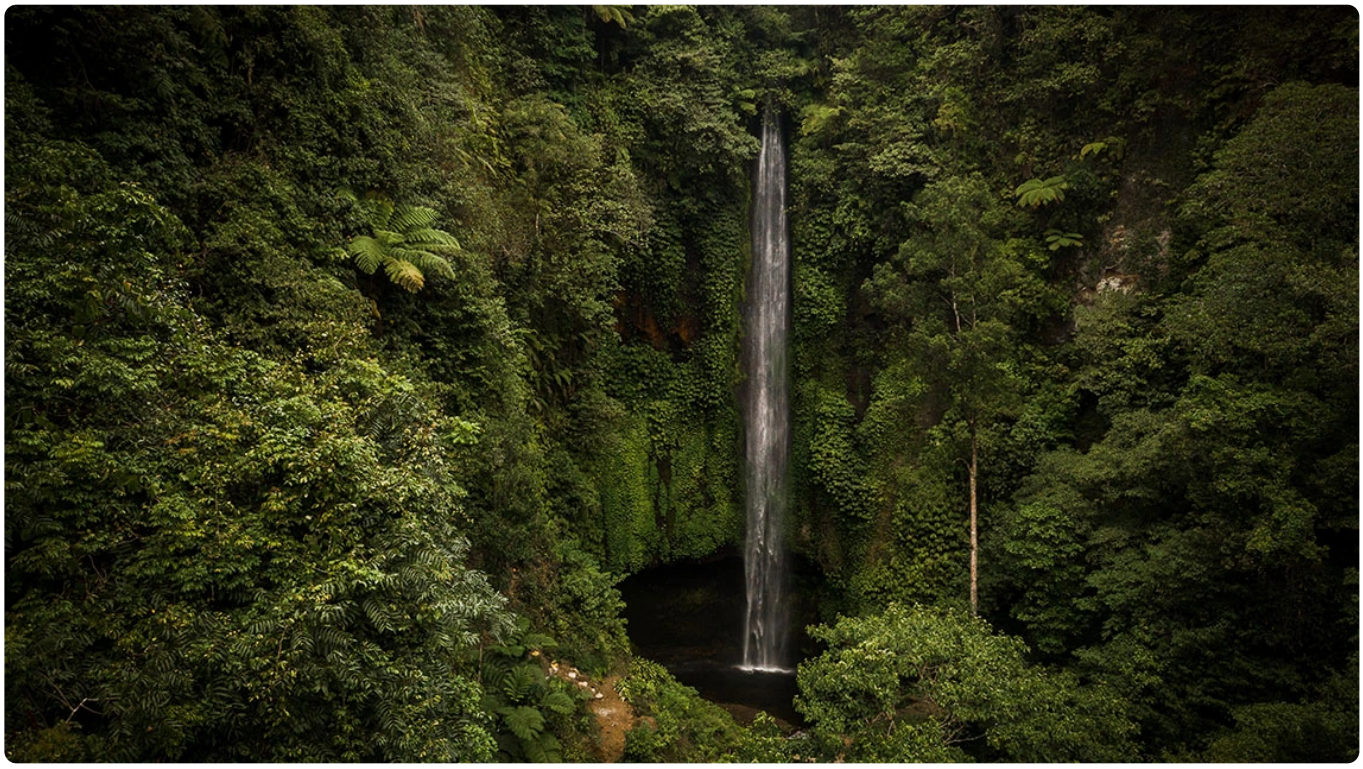
08｜回声消除算法实践指南

2021-12-08 冯建元

《搞定音频技术》 课程介绍 >



讲述：冯建元

时长 12:33 大小 11.50M

你好，我是建元。

上节课我们学习了回声消除算法的基本原理。我们看到，回声消除会受到声学环境、采集播放设备等多种因素的影响。因此，要想实现一个鲁棒、高效的回声消除算法是一件比较有挑战的事情。而在实际的音频实时互动场景中，回声问题可能也是我们碰到的最多的问题之一。

值得注意的是，音频处理往往是一环套一环的链路式的处理结构，回声消除作为音频前处理链路的一环很可能会对整体的音频体验产生影响。所以这节课我会先带你从整体上，了解一下实时音频互动系统的链路是怎么搭建的。然后我们再从几个案例出发，看看我们是如何改进回声消除算法，以及和其它模块配合，来整体提升实时音频的质量的。

实时音频处理链路

从贝尔实验室发明电话后，实时音频处理的技术不断发展，至今基于 VoIP（Voice over Internet Protocol，基于 IP 的语音传输）的实时音频互动已经慢慢成为人们日常音频在线交流的主流方法。图 1 展示的就是这样一个“现代版”的基于 VoIP 的实时音频处理链路。

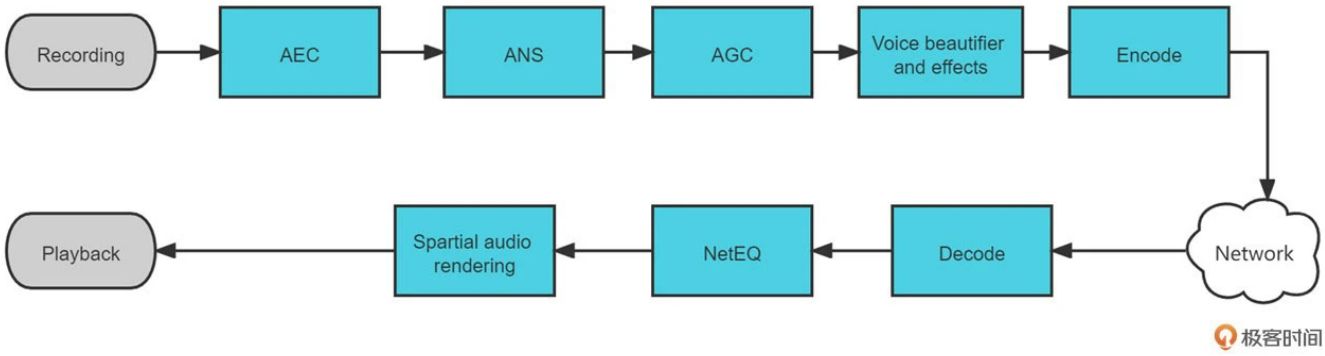


图1 实时音频处理链路

实时音频可以分为上行链路和下行链路两个部分，其中**上行链路**中的音频处理步骤可以对应图 1 中的上半部分。我们可以看到，在采集模块（Recording）之后就是我们常说的**“3A”处理，包括了 AEC（回声消除）、ANS（降噪）、AGC（音频增益控制）这 3 个部分。**之后在 Voice Beautifier and Effects（美声音效）模块我们可能会对人声做一些美化或者加入一些音效，来提升音频的听感。最后把音频进行编码（Encode）传输，至此上行链路也就结束了。

下行链路主要的流程是先从网络中获取音频包进行解码（Decode）。由于网路传输过程中可能会出现网络抖动、丢包等现象，而下行链路中的 NetEQ 模块就是用来解决这些网络带来的问题。随后在播放端可能会需要一些个性化的空间音频渲染（Spartial Audio Rendering ），把音频转换成符合空间听感的多声道音频。最后通过播放模块（Playback）将音频通过播放设备播放出来。

好的，了解了实时音频的处理步骤，我们再聚焦下 AEC 的位置。AEC 的位置放在紧挨着采集模块的地方。上节课我们介绍过 **AEC 需要对回声路径做估计，如果有别的模块放在 AEC 之前就会导致回声路径无法收敛到正确的位置。**比如将 AGC 放在 AEC 之前，那么 AGC 模块导致的音量波动就可能导致回声信号忽大忽小，从而 AEC 中滤波器的系数就很难收敛了。但这样的顺序安排是否就没问题了呢？

在链路式的处理方法中上游的算法会对下游的算法产生影响。AEC 在回声消除的时候如果对近端的声音也造成了损伤，比如，近端的部分噪声被当做回声消除了。之前降噪的部分我们讲过降噪模块由于也需要输入信号中的噪声来做噪声估计，如果输入信号中的噪声不是连续的（有部分被 AEC 干掉了），那降噪模块可能就会估计出一个偏小的噪声。这种情况下，回声虽然没有泄漏，但噪声的残余却增加了。

另外，回声消除的远端信号实际上需要的是扬声器播放的信号，也就是说需要在靠近 Playback 模块之前获取。如果信号在送往扬声器之后被系统再处理一遍（譬如动态范围控制，简称 DRC），就会干扰到 AEC 的效果。

所以音频的体验是一个需要整体考虑的问题，我们在音频模块的改进和修改的时候一定要对音频链路有一个整体的认识和理解，不然很可能会出现一些意想不到的体验问题。

回声消除算法案例分析

下面我们就结合 2 个常见的案例来分析分析，了解一下回声消除算法可能出现的问题以及回声消除算法的一些改进方向。

案例 1：混响

现在办公室的房间里有很多周围是玻璃墙的会议室，而在这些房间里回声泄漏的现象出现的概率很高。这是为什么呢？这里需要一点声学知识，即光滑且表面坚硬的墙壁，声波在其表面反射后能量衰减较小。也就是说声波可以在房间墙面多次反射，而能量依旧不会完全衰减。那这些**不断反射的声音就形成了一个持续时间很长的混响**，也就是所谓的“余音绕梁”。

我们知道，在回声消除中自适应滤波器中滤波器的长度是固定的。如果混响持续的时间超过了滤波器的长度，那么回声路径就不能被完整估计出来，从而就会导致回声泄漏。这种情况下，你可能会想，如果我们发现这些玻璃房间的混响时间太长，是不是把自适应滤波器的长度拉长就可以解决问题了呢？回答这个问题之前，我们先看一下下面的混响衰减曲线图。

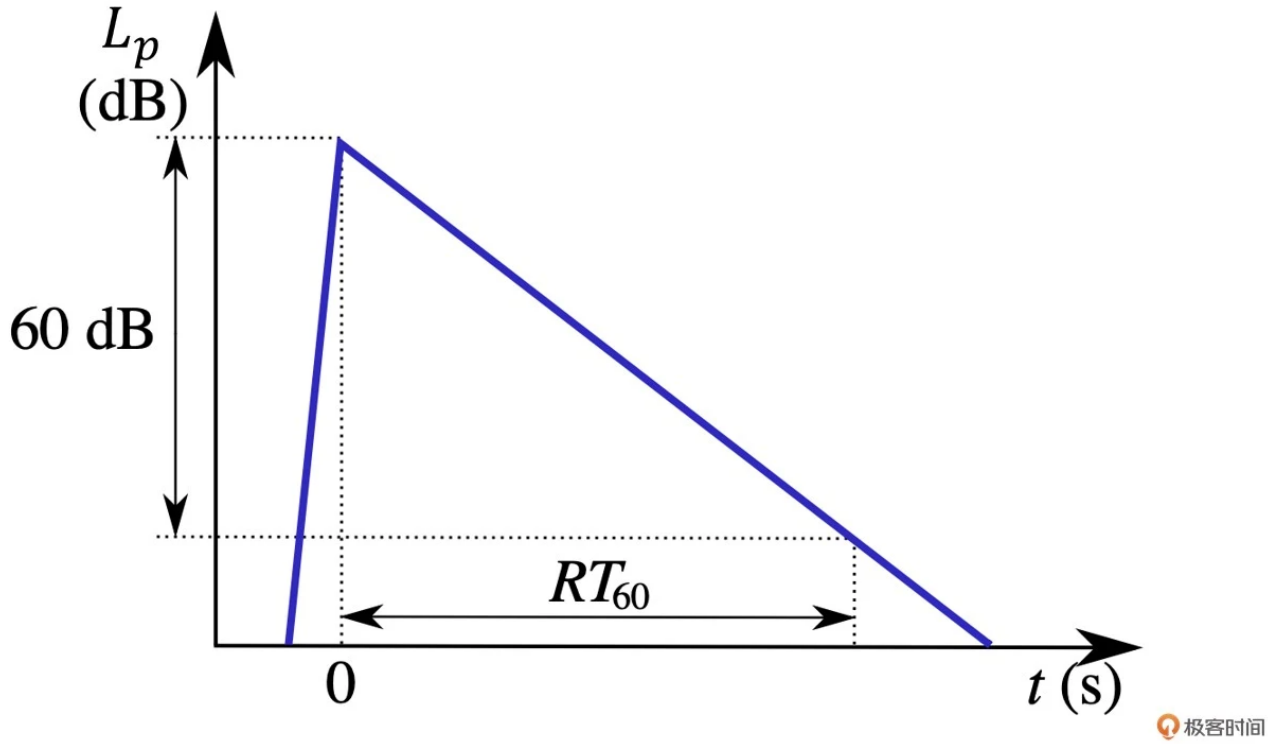


图2 混响衰减曲线图

如图 2 所示，**混响时间我们一般用 RT60 来表示**，也就是房间混响（Reverberation Time）衰减 60dB 所需要的时间。普通房间的混响的 RT60 大约在 50～200ms 左右，但有的玻璃房间或者大房间的 RT60 可能高达 200ms 甚至更多。假设我们音频的采样率是 16kHz，也就是说我们的自适应滤波器至少需要 3200 的长度，如果我们直接使用 NLMS 来计算，那可能实时性就没有办法得到满足。

为了克服滤波器长度太长造成的算力过大的问题，我们一般用分块频域自适应滤波器（Partitioned Block Frequency Domain Adaptive Filter， 简称 PBFDAF）来解决。**自适应滤波器耗时最长的就是卷积计算**。而**卷积计算是可以采用 FFT 来进行加速计算**的。

PBFDAF 的思想就是先把滤波器的权重和输入向量都分成等长的多块，然后通过 FFT 变换到频域来进行频域滤波，之后再通过 IFFT 得到滤波后的时域信号，最后把所有小块的结果叠加起来就可以得到滤波器的输出向量了。

卷积的算力是 $O(n^2)$，而 FFT 的算力是 $O(nlog(n))$，这样就实现了卷积计算的加速。实际上现在主流的开源算法像 WebRTC 和 Speex，在计算 NLMS 时都是采用类似这样的频域计算来对 NLMS 进行加速的。

好的，回到我们之前的问题，即混响情况下，如果滤波器长度不够就会导致回声泄漏，但是自适应滤波器的长度太长也会导致收敛缓慢。就算是使用了 PBFDAF 做加速，滤波器的

长度也不能随意设置。因此，在实际中我们可以先估算一下场景中 RT60 的分布。

例如，一般办公场景中正常办公室 RT60 很少超过 200ms，但可能会经常出现需要快速收敛的场景，比如房间的切换，那么滤波器可以短一些。但一些课程直播或者娱乐场景中场地有很长的 RT60 时间，但场景一般不会切换，这时则可以把滤波器长度调整得稍微长一些。

案例 2：非线性

Speex 作为较早开源的实时音频库，有不少实时音频系统都在使用它。但我们在实际使用的过程中会发现，**Speex 在一些采集播放设备的失真比较大的时候效果就比较差了**。

在上一讲中我们知道 NLMS 其实只能解决线性部分的变化，而扬声器、麦克风导致的非线性变化则可能需要非线性处理（Non-linear Process，NLP）来帮助解决。而 **Speex 并没有 NLP 模块来处理这种非线性的回声残留**，这也正是 Speex 的效果在不同设备上表现不稳定的原因。

相比较于 Speex 这种主要为 DSP 设备服务的算法，WebRTC 中的 AEC 由于考虑到要适配不同的设备终端，所以相比于 Speex，其主要增加了延迟估计和非线性处理这两个步骤来提高 AEC 算法的鲁棒性。那么 WebRTC 中 AEC 算法里的非线性处理是怎么做的呢？

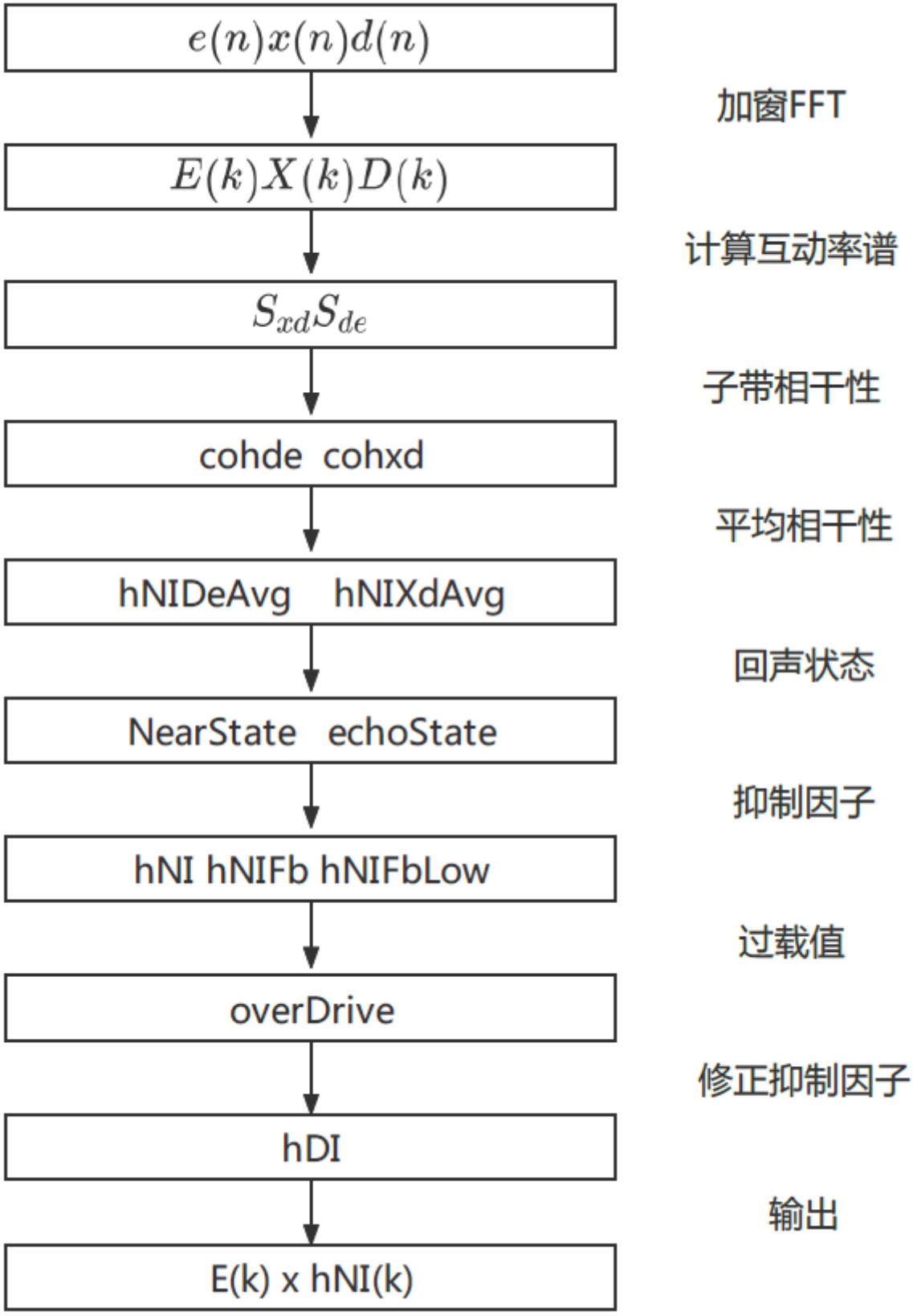


图3 WebRTC中的NLP处理（实时语音处理实践指南 中国工信出版社）

WebRTC AEC 代码中的 NLP 的做法如图 3 所示，主要是利用信号之间的相干性来做回声抑制的判断。这里的相干性主要是指近端信号 $d(n)$ 和残差信号 $e(n)$ 之间的相干性 c_{de}，以及近端信号 $d(n)$ 和远端信号 $x(n)$ 之间的相干性 c_{dx}。

其实这很容易理解：如果 c_{de} 越大，则代表近端信号和残差信号相似度高，也就是说回声很小，那么就越不需要去抑制；而如果 c_{dx} 越大，则代表近端信号和远端信号很相近，需要更多地抑制回声。

注意，这里由于非线性部分的回声可能是出现在某些频段上的，所以需要先把频带分解成多个的子带，且在每个子带上都做一个相干性的判断，然后再综合各个子带的相干性来计算回声抑制因子。而根据不同的回声状态，比如双讲情况下线性部分回声泄漏比较大，则需要增加一些 overDrive 的操作。

比如，本来可能算出来需要抑制 50% 的频谱能量，但由于判断此时是回声泄漏比较多的场景，这里再多做个 0.2 倍，变成抑制 60%。这里的 overDrive 就是作为经验值进一步防止回声泄漏。

值得注意的是，WebRTC 中 NLP 的处理在双讲情况下很容易把近端的噪声也一起干掉了，从而会影响到后续降噪模块的处理。所以我们一般会在 NLP 之后再补一个舒适噪声，用来保证降噪模块中噪声估计不会受到太大的影响。

小结

好的，现在我们来总结一下这节课的内容。回声消除作为实时音频处理链路中的重要的组成部分，和其它模块的表现会相互耦合，并且每个模块摆放的位置、处理的方法都可能对实时音频的整体体验产生影响。而房间的混响、非线性处理等方面都是我们可以改进回声消除效果的方向。

其实Speex和WebRTC 的开源已经让整个 AEC 的发展前进了一大步。利用起来后就再也不用从零开始搭建回声消除算法了。Speex 主要采用了前景滤波器和背景滤波器的双滤波器结构，这样的结构能更好地防止滤波器发散，使得在处理双讲时效果更鲁棒。但是 Speex 没有 NLP 部分的处理，所以在面对非线性问题时就会捉襟见肘。

比较可惜的是 Speex 目前已经停止更新，所以一般基于 Speex 的项目都会自行再做一些非线性的处理。而 WebRTC 的线性部分只有单路滤波器，较 Speex 略弱，所以主要靠 NLP 部分的处理来提升整体效果。但其实，最近更新的 WebRTC aec3 的算法中也使用了基于 NLMS 和卡尔曼滤波，双滤波器的结构，这使得线性部分能够得到进一步的加强。

另外，我们看到非线性部分还需要一些人工的经验数值来做一些状态判断和 overDrive 等操作。而实际上基于传统算法的 AEC 很难彻底解决由于环境、设备等时变性、非线形导致的不稳定的问题。

因此，最近几年基于机器学习的方法被用到了 AEC 的领域。尤其是在 NLP 的部分，和之前讲的降噪模型类似，基于机器学习的模型可以把 AEC 线性部分的结果作为输入，结合近端和远端的信号自动进行残余回声的消除，这里你可以参考一下最近举办的 AEC challenge的比赛，里面有很多优秀的论文和数据可以参考和使用。

思考题

这里给你留个思考题。回声消除其实还是不可避免的会对音质造成一些损伤，那么请你想一想如果是音乐场景，我们在做 AEC 的时候可以做哪些改进来提升音质呢？提示：音乐场景的音频采样率会比较高。

欢迎留言和我分享你的思考和疑惑，你也可以把今天所学分享给身边的朋友，邀请他加入探讨，共同进步。下节课再见！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 07 | 如何通过算法自动快速地消除回声？

下一篇 09 | 音频编解码器是如何工作的？

精选留言 (3)

写留言

徐刚

2021-12-08

老师，推荐下nlp的AI处理可以参考哪些论文，或开源代码？

Geek_7de4c5
2021-12-08

老师双讲检测有哪些论文可以推荐一下吗？感谢

Geek_7de4c5
2021-12-08

音乐场景下能量集中在高频，尽量避免做对高频做非线性处理。